FICHE 42 : *Numération et codage*

I. Les bases de numération :

Ce système de numération, utilisé dans la vie quotidienne, dispose de 10 symboles différents : 0, 1, 2, 3, 4, 5, 6, 7, 8 et 9. On parle de base 10.

Un nombre entier positif N s'écrit en base 10 :

$$(N)_{10} = a_n \cdot 10^n + a_{n-1} \cdot 10^{n-1} + \ldots + a_2 \cdot 10^2 + a_1 \cdot 10^1 + a_0 \cdot 10^0 = a_n a_{n-1} \ldots a_2 a_1 a_0$$

Pour spécifier la base utilisée, on place généralement le nombre entier positif entre parenthèses, suivi de la base utilisée en indice.

Exemple :

$$(7\ 239)10 = 7\ 239 = 7 \cdot 10^3 + 2 \cdot 10^2 + 3 \cdot 10^1 + 9 \cdot 10^0$$

Pour la base 10, on oublie souvent de spécifier la base car il s'agit de la base de numération utilisée quotidiennement.

Les **digits** correspondent aux coefficients a_n. Ils ne peuvent prendre que des valeurs appartenant à la base. Les **poids** sont les puissances de 10. Le poids est égal à la base élevée à la puissance de son rang.

	Unité	Dizaine	Centaine	Millier	10 Milliers	100 Milliers
Digit	a_0	a_1	a_2	a_3	a_4	a_5
Rang	0	1	2	3	4	5
Poids	10^0	10^1	10^2	10^3	10^4	10^5

Pour faciliter l'écriture, les représentations suivantes sont possibles :

$$(N)_{10} = N = (a_n a_{n-1} \cdots a_1 a_0)_{10} = (a_n a_{n-1} \cdots a_1 a_0)$$

II. Numération binaire (base 2) :

a. La numération binaire ou en base 2

La numération binaire ou en base 2 utilise deux symboles : 0 et 1. Cette base est très commode pour distinguer les deux états logiques fondamentaux (lampe allumée ou éteinte, présence ou non dans une pièce, couleur noir ou blanc…), et est très utilisée en informatique.

On écrit :

$$(a_{n-1} a_{n-2} \cdots a_1 a_0)_2$$

Exemple :

$$(100110000101)_2$$

En numération binaire, les digits (a_i) sont appelés **bit** (abréviation de Binary digIT).
Un code binaire (à n bits en base 2) distingue 2^n états ou combinaisons.

n	0	1	2	3	4	5	6	7	8
$\mathbf{2^n}$	1	2	4	8	16	32	64	128	256

- Les puissances successives de 2 (1, 2, 4, 8, 16, 32…) sont appelées **poids binaires**.
- Le bit de poids le plus fort (a_{n-1}) est appelé **MSB (Most Significant Bit)**.

Ressources

- Le bit de poids le plus faible (a0) est appelé **LSB (Low Significant Bit)**.
- Un regroupement successif de 4 bits s'appelle un **quartet**.
- Un regroupement successif de 8 bits s'appelle un **octet**.
- Un regroupement successif de k bits (k > 8) s'appelle un **mot de k bits**.

b. Numération hexadécimale (base 16)

Ce système de numération est très utilisé dans les systèmes ordinateurs et micro-ordinateurs ainsi que dans le domaine des transmissions de données.

Il comporte 16 symboles : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

Un nombre entier positif N s'écrit en base 16 :

$$(N)16 = (a_{n-1}\ a_{n-2}\ \dots\ a_1\ a_0)_{16}$$

Pour indiquer la base 16, on peut utiliser le caractère $ (dollar) devant le nombre, ou alors 16# devant le nombre.

Exemple :

$$(A8)_{16} = 16\#A8 = \$A8$$

III. Les changements de base

a. Conversions directes

- Du binaire vers l'hexadécimal

Pour convertir du binaire vers l'hexadécimal, on divise le nombre binaire en quartet, en partant de la droite. Chacun des paquets est ensuite converti en hexadécimal.

Exemple :

$$(110101110001)_2=(11010111\ 0001)_{16}=16\#D71$$

- De l'hexadécimal vers le binaire

C'est le processus directement inverse, on écrit chaque quartet sur 4 bits en binaire en complétant éventuellement avec des zéros sur la gauche.

Exemple :

$$1C35H = (0001\ 1100\ 00110101)_2$$

b. Conversions indirectes

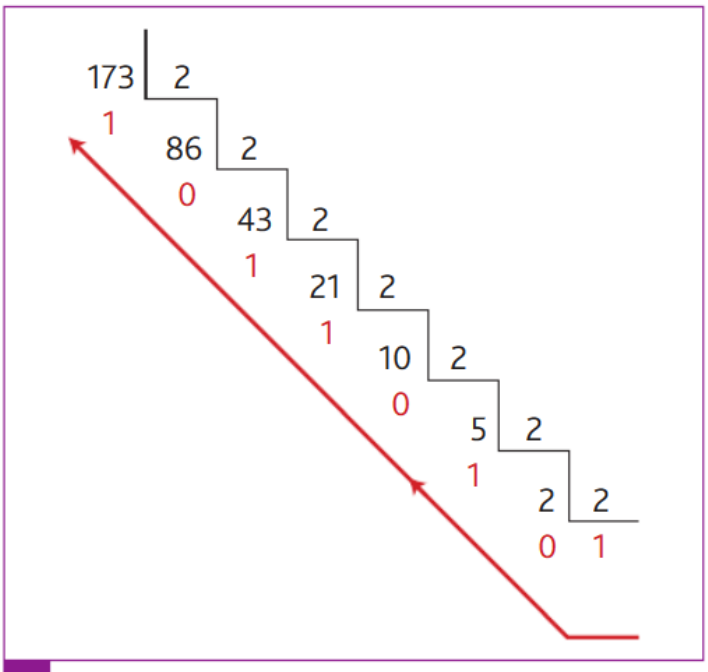
Un nombre entier positif N étant donné en base 10, on cherche à l'écrire dans une autre base notée b. La méthode consiste à diviser le nombre décimal N et à conserver le reste (division entière). Le quotient obtenu est ainsi successivement divisé tant qu'il n'est pas nul. Les restes successifs sont écrits, en commençant par le dernier, de la gauche vers la droite, pour former l'expression de N dans le système de base b.

- Du décimal vers le binaire

Exemple pour la conversion de 173 en base 2 : le résultat est donc $(10101101)_2$ [document 11].

- Du décimal vers l'hexadécimal

On reprend la même méthode mais en divisant par 16.



11 Conversion de 173 en base 2.

Ressources

IV. Le codage des nombres

a. Codages pondérés

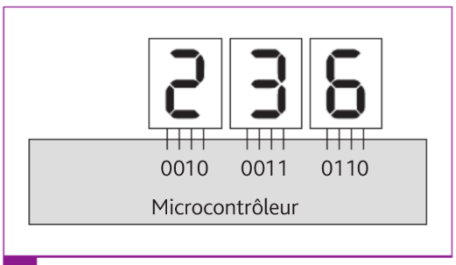
Les codes pondérés sont des codes où **chaque digit est affecté d'un poids**.

- **Les codes naturels**

Les codes binaires, décimaux et hexadécimaux répondent aux règles classiques de l'arithmétique des codes pondérés ; ce sont donc des codes pondérés.

- **Le Code Binaire Codé décimal (BCD)**

Ce codage est destiné à l'affichage de valeurs décimales, chaque digit (unités, dizaines, centaines…) doit être codé en binaire sur 4 bits. Il ne permet aucun calcul, il est uniquement destiné à la saisie et à l'affichage de données.



12 Utilisation du codage BCD.

Exemple :

$$(7239)_{16} = (\underbrace{0111}_{7_{(2)}}\ \underbrace{0010}_{2_{(2)}}\ \underbrace{0011}_{3_{(2)}}\ \underbrace{1001}_{9_{(2)}})_{BCD}$$

- **Le code complément à 1**

Le code complément à 1, appelé aussi Complément Restreint (CR) d'un nombre, est obtenu en complémentant chaque bit un à un.

Exemple : $CR(22_{10}) = CR(10110_2) = (01001)$

- **Le code complément à 2 :** représentation des nombres négatifs

Le code complément à 2, appelé aussi complément vrai (CV), permet de représenter les nombres entiers négatifs utilisés dans les calculateurs. Il est obtenu par :

$$-x = CV(x) = CR(x) + 1$$

Le signe est le bit de poids fort (MSB). Par convention, 0 pour une valeur positive et 1 pour une valeur négative.

Exemple : -22 = CV(2210) = CV (0101102) = CR (0101102) + 1 = (101001) +1=1101010

b. Codages non pondérés

Les codes non pondérés correspondent aux codes où **chaque digit n'a pas de poids**.

- **• Le code binaire réfléchi ou code Gray**

Dans ce code, un seul bit change entre 2 valeurs adjacentes. Le bit changeant est celui le plus à droite ne provoquant pas une combinaison déjà apparue. Il est employé dès que l'on doit représenter une évolution réelle des variables où une seule change d'état à un instant donné.

V. L'extension aux codes non numériques

Pour manipuler d'autres éléments que des nombres, il est aussi nécessaire de les coder. Le plus connu de ces codes, et le plus utilisé, en particulier dans le monde informatique, est le **code ASCII** (American Standard Code for Information Interchange) [documents 13 et 14]. Certains codes ne sont quasiment plus utilisés à l'exception de SOH, STX, ETX, EOT, LF et CR [document 14].

Ces codes alphanumériques sont accessibles sur un ordinateur fixe avec un clavier ou depuis un ordinateur portable disposant d'un pavé numérique. Pour cela, ouvrir un éditeur de texte, appuyer sur la touche « Alt » et saisir un nombre compris entre 32 et 127 sur le clavier numérique. Le caractère associé au code ASCII apparaîtra.

Ressources

Déc	Hex	Char	Déc	Hex	Char	Déc	Hex	Char	Déc	Hex	Char
0	00	Null	32	20	Space	64	40	@	96	60	`
1	01	Start of header	33	21	!	65	41	A	97	61	a
2	02	Start of text	34	22	"	66	42	B	98	62	b
3	03	End of text	35	23	#	67	43	C	99	63	c
4	04	End of transmit	36	24	$	68	44	D	100	64	d
5	05	Enquiry	37	25	%	69	45	E	101	65	e
6	06	Acknowledge	38	26	&	70	46	F	102	66	f
7	07	Audible bell	39	27	'	71	47	G	103	67	g
8	08	Backspace	40	28	(	72	48	H	104	68	h
9	09	Horizontal tab	41	29	)	73	49	I	105	69	i
10	0A	Line feed	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	43	2B	+	75	4B	K	107	6B	k
12	0C	Form fed	44	2C	’	76	4C	L	108	6C	l
13	0D	Carriage return	45	2D	–	77	4D	M	109	6D	m
14	0E	Shift out	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	47	2F	/	79	4F	O	111	6F	o
16	10	Data link escape	48	30	0	80	50	P	112	70	p
17	11	Device control 1	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	50	32	2	82	52	R	114	72	r
19	13	Device control 3	51	33	3	83	53	S	115	73	s
20	14	Device control 4	52	34	4	84	54	T	116	74	t
21	15	Neg. aknowledgement	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	54	36	6	86	56	V	118	76	v
23	17	End trans. block	55	37	7	87	57	W	119	77	w
24	18	Cancel	56	38	8	88	58	X	120	78	x
25	19	End of medium	57	39	9	89	59	Y	121	79	y
26	1A	Substitution	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	59	3B	;	91	5B	[	123	7B	{
28	1C	File separator	60	3C	<	92	5C	\	124	7C	|
29	1D	Group separator	61	3D	=	93	5D	]	125	7D	}
30	1E	Record separator	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	63	3F	?	95	5F	_	127	7F	□

13 Tableau de codage décimal/hexadécimal/ASCII.

ASCII	Caract.	Signification	ASCII	Caract.	Signification
00	NUL	*null*, nul	016	DLE	*data link escape*, échap. liaison données
01	SOH	*start of header*, début d'en-tête	017	DC1	*device control 1*, commande unité 1
02	STX	*start of text*, début de texte	018	DC2	*device control 2*, commande unité 2
03	ETX	*end of text*, fin de texte	019	DC3	*device control 3*, commande unité 3
04	EOT	*end of transmission*, fin de transmission	020	DC4	*device control 4*, commande unité 4
05	ENQ	*enquiry*, interrogation	021	NAK	*negative acknowledgement*, acc. récep. nég.
06	ACK	*acknowledge*, accusé de réception	022	SYN	*synchronous idle*, inactif synchronisé
07	BEL	*bell*, sonnerie	023	ETB	*end of transmission block*, fin tran. bloc
08	BS	*backspace*, espacement arrière	024	CAN	*cancel*, annuler
09	HT	*horizontal tabulation*, tabulation horiz.	025	EM	*end of medium*, fin du support
010	LF	*line feed*, saut de ligne	026	SUB	*substitute*, substitut
011	VT	*vertical tabulation*, tabulation verticale	027	ESC	*escape*, échappement
012	FF	*form fed*, saut de page	028	FS	*file separator*, séparateur de fichiers
013	CR	*carriage return*, retour chariot	029	GS	*group separator*, séparateur de groupes
014	SO	*shift out*, hors code	030	RS	*record separator*, sép. d'enregistr.
015	SI	*shift in*, en code	031	US	*unit separator*, séparateur d'unités

14 Tableau de signification du codage.

Ressources